# Implementation of Load–Balancing and Auto-Scaling on Oracle Virtual Box

**Preksha Sundrawat[1], Meghna Sharma[2], P.K Bishnoi[3]**
[1, 2, 3] Mody University of Science and Technology

***Abstract-*** *The key objective of this research paper is to design and realize the model for load balancing and auto scaling by using Oracle Virtual Box. Load balancing and auto scaling are the elementary key activities of Cloud Computing. Virtual Box is being used as software of virtualization in this implementation. Auto-scaling and load balancing will be applied on the virtual machines of Oracle Virtual Box.*

*Java programs will be checking the utilization of virtual machines, on the basis of that load will be distributed and new clone will be created at real time. New clone/virtual machine will be same in terms of software, data and OS but will be different in terms of IP address and other machine identities. Both the important features of cloud computing, to improve the availability and reliability of resources and services on cloud for programmers and end users.*

***Keywords***- Auto-scaling, load balancing and virtualization.

## I. INTRODUCTION

Cloud means any service that is on a virtual network for example like many organization which could be smaller, medium that provides user with services to the end user. These organizations have services and centres like data centres like they used traditional methodology for mounting and accessing these service database server for mail server and many more.

### 1.1 Service Models of Cloud Computing:

Cloud computing is based on service models, the word service here reflects the concept of reusability of cloud component across the service provider's network, and they are namely as:

1.1.1 Infrastructure-as-a-Service (IaaS): It is the layer of cloud computing that provides the user with the infrastructure in terms of servers, networking, storage, processing etc. It is a pay per model (e.g., Amazon Web Services).

1.1.2 Platform-as-a-Service (PaaS): We are provided a platform basically and solely for developing various kinds of application that is required to run or provide the services from the particular application (e.g., Google App Engine, Microsoft Azure).

1.1.3 Software-as-a-Service (SaaS): This layer provides the host with all the software service over Internet. (e.g., Web Based email, Google Docs).

### 1.2 Deployment Models of Cloud Computing:

After the service model, there come the different types of deployment models which are:

1.2.1 Public Cloud: This cloud can be accessed over the internet for the services. Anyone can access the services and resources which are hosted by the public cloud.

1.2.2 Private Cloud: It allows us to use the software services within a defined premise and is managed by third party.

1.2.3 Hybrid Cloud: It is a combination of private and public cloud, when these two clouds can't meet up their requirements for an organization these two clouds are put out and they are integrated into a new cloud called as hybrid cloud.

1.2.4 Community Cloud: It is a type of cloud that is shared among various organizations with a common tie managed by Third Party Service (TPS) and made available to on an off premise.

The characteristics of cloud has made it most demanding technology in IT field The major characteristics are on demand service, pay as u go , distributive service made it popular and apart from this pooled resource, real network use and elasticity made it easier to use .

### 1.3 Cloud Scalability:

Scalability is the capacity of handling a growing amount of work in an accomplished manner. It is an ability for a system network or a process to handle growing amount of work. The word scalability could be used in context of hardware system as well as the program.

In order for a system to handle more work one of the two things should happen i.e. either we need to give more

power to that system or we need to add more system. So scalability can be attained through one of the two methods-

1.3.1 Vertical Scalability: It is also known as scaling up which means we add more resources to the existing computer or node. For example we might want to increase the size or add more CPUs, memory or hard disks.

No matter how scalable or powerful our computer is, vertical scalability has a limit i.e. we cannot make a single computer more powerful than a particular point so this is where the traditional approach of handling big data fails.

1.3.2 Horizontal Scalability: Here, rather than making a node more powerful we add another node so we keep on adding more and more node as we need more and more power. The term scaling out is also used for horizontal scaling; there is usually a linear correlation between the number of computer and performances. It works on a distributed system model.

**1.4 Auto Scaling:** It is a technique through which it automatically scales up or down the capacity without any user defined condition. The auto scaling in cloud infrastructure is shown below:

Auto scaling has wonderful features, some of them are:

- When demand increases it automatically scales out the instances.
- Scales out the unneeded virtual machine instances automatically and save money when demand decreases.
- Replace unused instances to maintain higher availability of our virtual machine for use.

**1.5 Load Balancing:**

It is the technique through which the workload is divided among the different nodes ensuring that no node in the system is overloaded or sitting idle at any instant of time. An effective load balancing algorithm will make sure that each and every node of the system has averagely the same amount workload. The load balancer checks the workload of each and every node in the system and if it founds that the load of a particular node is extremely more than the other, it transfers the load to the other. Also if the load on the entire load is averagely same and high, it generates a new virtual machine and transfer the load to the new machine.

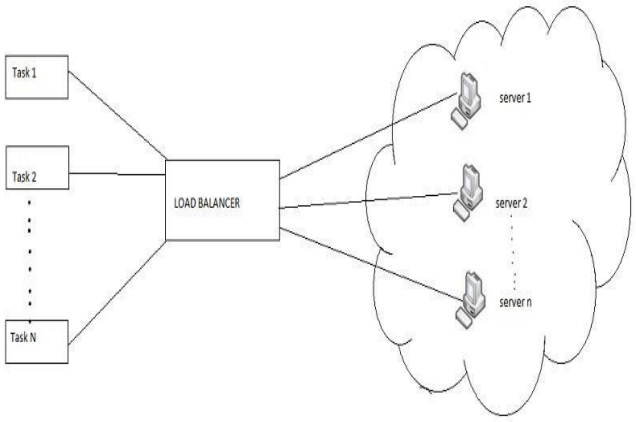


Fig 1: Load Balancer

## II. REVIEW OF OTHER PAPERS

[1] V. Shah; H. Trivedi says in paper "A distributed dynamic and customized load balancing algorithm for virtual instances" that Cloud computing is one of the most prominent technologies. The fundamental idea behind cloud computing is to distribute an array of computing services by unifying and scheduling a pool of computing resources, thereby minimizing the burden on the users and helping them focus on their core businesses. These computing resources are hosted on virtual hosts and distributed on-demand to the users by cloud service providers. For efficient resource utilization, systematic load balancing of incoming user traffic across virtual hosts is imperative.

[2] A. Anwar; A. Sailer; A. Kochut; C. O. Schulz; A. Segal; A. R. Butt says in paper Scalable Metering for an Affordable IT Cloud Service Management that As the cloud services journey through their life-cycle towards commodities, cloud service providers have to carefully choose the metering and rating tools and scale their infrastructure to effectively process the collected metering data. In this paper, we focus on the metering and rating aspects of the revenue management and their adaptability to business and operational changes. Our analysis shows that service management related tasks can be offloaded to the existing VMs with at most 15% overhead in CPU utilization, 10% overhead for memory usage, and negligible overhead for I/O and network usage.

[3] E. Campos; R. Matos; P. Maciel; I. Costa; F. A. Silva; F. Souza says in paper Performance Evaluation of Virtual Machines Instantiation in a Private Cloud that Elasticity is an outstanding concept of cloud computing, usually deployed through mechanisms such as auto scaling and load balancing. Cloud-based applications are able to adapt themselves dynamically to the workload behavior due to such mechanisms. The efficient instantiation of Virtual Machines

(VMs) is one requirement for the elastic behavior of cloud-based applications. The model evaluation showed that for 6 GB and 8 GB MI, the probability of finding the MI on cache must be at least 40 % and 60 % respectively, to achieve an average instantiation time of 300 seconds. For MI with size 2 GB, such time is not exceeded even with the cache disabled.

[4] R. Poddar; A. Vishnoi; V. Mann says in paper HAVEN: Holistic load balancing and auto scaling in the cloud that Load balancing and auto scaling are important services in the cloud. Traditionally, load balancing is achieved through either hardware or software appliances. Hardware appliances perform well but have several drawbacks. They are fairly expensive and are typically bought for managing peaks even if average volumes are 10% of peak. Further, they lack flexibility in terms of adding custom load balancing algorithms. They also lack multi-tenancy support. For this reason, HAVEN is able to provide performance at par with a hardware load balancer while still providing the flexibility and customizability of a software load balancer. We validate HAVEN on a hardware setup and our experiments confirm that it achieves high performance without any significant overdeads.

## III. DESIGN

PREREQUISITES:

- A requesting client.
- Server machine with JSP program.
- Program for new machine creation. This program resides in Host machine.
- Program for utilization checking. This program sides in each and every virtual machine.
- Two or more host machines on which virtual machines run.
- Virtual machine called as Guest Machines.

A database with attributes like memory utilization, process utilization, machine name. This database is having connectivity with JSP program.

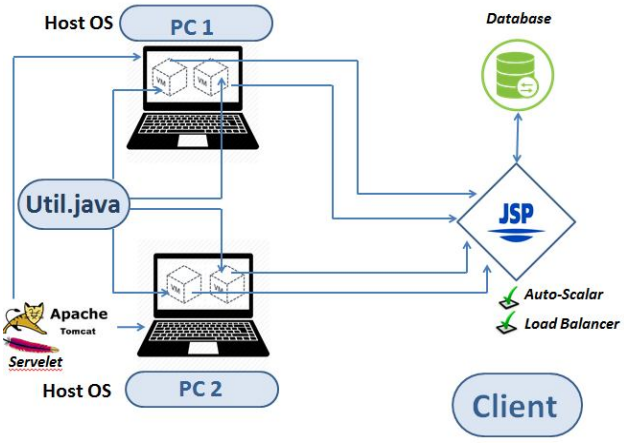


Fig 2: Design of the hardware

STEP1: In initial stage 2 or more virtual machines are running side by side on a oracle virtual box setup. This setup is on our host machine. There entries in database are registered as "Machine 1" and "Machine 2" with their respective memory and processor utilization.

STEP 2: With duration of time request for resources on currently working virtual machines may increase and they are unable to fulfill these requests due to overload. In such scenario following process in sequence takes place:

**PROCESS 1 LOAD BALANCING:** This fundamental feature of cloud computing is used to balance the load on all the running machines and check load in fix intervals of time. It checks the CPU utilization status of every clone machine .In our model we have implemented this thing by using a java program. This program is running on all the virtual machines. E.g.: Suppose two machines are currently in running phase .And **as** per our java program there CPU utilization is 20% and 30%. So job of load balancing is to equate this CPU utilization so that the results turns out to be 25% of each machine.

**PROCESS 2 AUTO SCALING:** As soon as load bearing capacity of running machine exceeds there upper limit , a process called machine creation which is running on our HOST machine comes into action and in real time creates clone of the virtual machines. This newly created virtual machine has same functionalities of main system. This entire procedure is called "AUTO SCALING".

**PROCESS 3 REGISTRATION IN DATABASE:** As previously mentioned for every machine entry is done in database. So as soon new machine is created through auto scaling its entry is done on the database and our database is updated. And this procedure goes on for every request of

client. Assembling all modules of our model a clear depiction of load balancing and auto scaling is achieved.

### 3.1 Algorithm:

```
FUNCTION 1()//Request  for resources from client
{
 JSP ()//server  for load balancer and auto scaling & all request come here
 { Database()
 int MU //Machine utilisation,
 int     PU//process utilization,
 int  MN //machine name;
MN++;
Printf("Enter machine utilization of new clone:" &MU)
Printf("Enter process utilization of new clone:" &PU)}

Load  compare()
{If (load 1==load 2)
{MU ++; PU++; MN++; }
Elseif(load1>load2)
{
    Load2++ }
Else{
Load1++ }

Machine capacity =10000;
If (mu1>1000 &&mu2>10000)
{
 CLONE  CREATION()//clone  is  created  of  same specifications,program in host machine.
{
  Database()
}
```

## IV. RESULTS

```
                                    CPU USAGE DETAILS
**** Sizes in Mega Bytes ****

PHYSICAL MEMORY DETAILS             Starting Test with 4 CPUs and random number:1687997611
                                    24.495535
                                    CPU USAGE : 24.0 %
total physical memory : 4043MB
total free physical memory : 1547MB
                                    CPU USAGE DETAILS
**** Sizes in Giga Bytes ****

DISC SPACE DETAILS                  Starting Test with 4 CPUs and random number:1160214953
                                    25.143442
                                    CPU USAGE : 25.0 %
Free Space in drive C: : 62.0GB
Free Space in drive D:  : 0.0GB
Free Space in drive E: 107.0GB      CPU USAGE DETAILS
 !!!alert!!!!

                                    Starting Test with 4 CPUs and random number:73371786
**MEMORY DETAILS  **                24.697847
                                    CPU USAGE : 24.0 %

Total Memory: 61MB
 Memory Used: 0MB                   CPU USAGE DETAILS
 Memory Free: 60MB
Percent Used: 1.5611477983676328    Starting Test with 4 CPUs and random number:213591692
Percent Free: 98.43885220163237%    25.55694
                                    CPU USAGE : 25.0 %
```

Fig 3: CPU utilization of the system

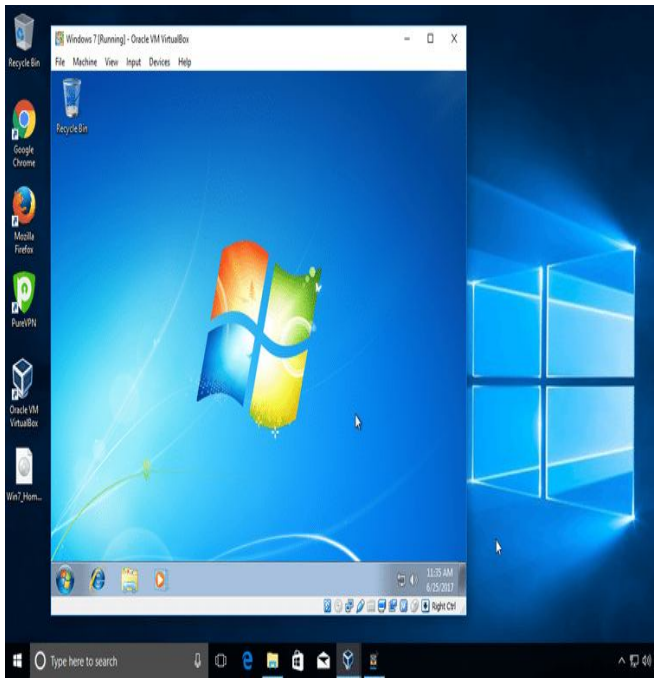


Fig 4: Creation of clone virtual machine in Virtual Box

### 4.1 Advantages:

4.1.1. Cost Efficient: We can work on the multiple operating systems without actually loading it in our system.

4.1.2. Storage Optimization: Using multiple operating systems on single machine saves the space.

4.1.3. Time saving: Deploying machine every time in hardware domain takes huge time unlike software deployment which could be implemented in real time

**4.2 Disadvantages:**

4.2.1. Non Synchronized: This model will generate virtual machine with similar version of operation system. The client may need a different operating system for his work, which is not provided.

4.2.2. Technical Issues: We need a system with large hard disk space and a fast processor or else creation of multiple virtual machines may lead to system crash or provide a low speed of performance.

4.2.3. Security: The major problem that comes into the picture is security. All the information that is being stored in the virtual machines is vulnerable to external attack or not secure.

## V. CONCLUSION

In this paper we discussed the procedure to implement auto scaling and load balancing using virtual box .In these work we have used oracle virtual box to create clone ,programs in java language have been used to keep the check on the CPU utilization of virtual machines. We created a prototype of cloud computing working on platforms like AWS. Load balancing and auto scaling are fundamentals of cloud computing. It is required to distribute the dynamic workload evenly across all the nodes to achieve a high user satisfaction and re-source utilization ratio by making sure that every computing re-source is distributed efficiently and fairly. This task is accomplished by load balancing and auto scaling.

## REFERENCES

[1] V. Shah; H. Trivedi,A distributed dynamic and customized load balancing algorithm for virtual instances,Department of Computer Engineering, Gujarat Technological University, Ahmedabad, India,pp. 1-6,26-28 Nov. 2015

[2] P. C. Amogh; G. Veeramachaneni; A. K. Rangisetti; B. R. Tamma; A. A. Franklin,A cloud native solution for dynamic auto scaling of MME in LTE,Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, India,pp. 1-7,8-13 Oct. 2017

[3] Y. T. Lee; H. L. Chao; J. W. Tang,Scalable and elastic cloud data center for self-organizing dense small cell networks,Institute of Computer Science, National Chiao Tung University, Hsinchu, Taiwan,pp. 420-423,19-21 Aug. 2015

[4] R. Poddar; A. Vishnoi; V. Mann,HAVEN: Holistic load balancing and auto scaling in the cloud,IBM Research, India,pp. 1-8,6-10 Jan. 2015